ATLAS
MANUAL DE USUARIO
COMPONENTE COMBO DE VALORES

Versión 1.3

Área de Aplicaciones Especiales y Arquitectura de Software

Agencia de Informática y Comunicaciones de la Comunidad de Madrid

Hoja de Control

Título	Manual de usuario Componente Combo de valores		
Documento de Referencia	NORMATIVA ATLAS		
Responsable	Área de Aplicaciones Especiales y Arquitectura de Software		
Versión	1.3	**Fecha Versión**	11/11/2013

Registro de Cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
1.0	Versión inicial del documento	Área de Integración y Arquitectura de Aplicaciones	16/03/2012
1.1	Apartado 4.5 – Añadido nuevo atributo externalProvider	Área de Integración y Arquitectura de Aplicaciones	24/10/2012
1.2	Añadida propiedad inputWidth	Área de Integración y Arquitectura de Aplicaciones	15/01/2013
1.3	Explicado que hay que insertar la fachada y que se puede usar el backing bean existente para guardar las propiedades de la lista de valores	Arquitectura de Aplicaciones	11/11/2013

Índice

1. INTRODUCCIÓN 4

1.1. AUDIENCIA OBJETIVO 4
1.2. CONOCIMIENTOS PREVIOS 4

2. DESCRIPCIÓN 4

3. INSTALACIÓN Y CONFIGURACIÓN 5

3.1. INSTALACIÓN 5
3.2. CONFIGURACIÓN 5

4. USO 7

4.1. PASO 1: DEFINICIÓN DE LA CONSULTA DEL COMBO DE VALORES 7
4.2. PASO 2: DECLARACIÓN EN EL BEAN DE RESPALDO DE LA PÁGINA DE UN OBJETO DE TIPO ATLAS.COMPONENTE.BO.COMBOVALORES E INICIALIZACIÓN DEL MISMO 7
4.3. PASO 3: DEFINICIÓN DEL ESPACIO DE NOMBRES DE ETIQUETAS DE ATLAS 9
4.4. PASO 4: INSERCIÓN EN LA PÁGINA DE LA ETIQUETA ATLAS:COMBOVALORES 10
4.5. ATRIBUTOS DEL COMPONENTE. 11
4.6. ETIQUETAS Y VALORES DEL COMBO. 15

5. RECOMENDACIONES Y BUENAS PRÁCTICAS 15

5.1. EJEMPLO DE USO 15

6. PREGUNTAS MÁS FRECUENTES 16

7. ENLACES RELACIONADOS 17

1. INTRODUCCIÓN

Este documento contiene el manual de uso del componente visual ***Combo de valores*** del Framework Atlas. En él se incluye información sobre cómo utilizar dicho componente en una aplicación Web, así como información acerca de la configuración de los parámetros fundamentales del componente.

1.1. AUDIENCIA OBJETIVO

Este documento está orientado a toda aquella persona que esté desarrollando una aplicación Web basada en el Framework Atlas y necesite utilizar componentes de presentación en su aplicación Web.

1.2. CONOCIMIENTOS PREVIOS

Para un completo entendimiento del documento, el lector deberá tener conocimientos previos sobre las siguientes tecnologías:

- Java Server Faces (JSF)
- Facelets
- Richfaces
- Spring Framework
- Hibernate

Para saber más sobre dichas tecnologías, consultar el apartado de este documento, *Enlaces Relacionados*.

2. DESCRIPCIÓN

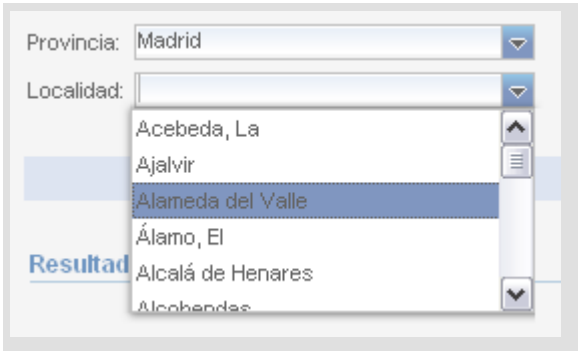
Este componente muestra al usuario un combo sobre el que seleccionar un valor de una lista. Tiene las siguiente características:

- La lista de elementos que muestra el combo se obtienen de una consulta SQL proporcionada por el usuario. También es posible especificar en la definición del componente una columna de ordenación y un filtro para la consulta. Este filtro puede ser dinámico haciéndolo depender de valores de un backing bean.

- Permite especificar si se quiere actualizar el valor cada vez que se modifique en el combo. Esta actualización se realiza mediante ajax, y en caso de actualizar, se puede especificar qué otros componentes deben procesarse y renderizarse.

- Permite escribir sobre el combo para filtrar valores y utilizar los cursores para seleccionar un valor concreto.

- Se puede limitar el número de elementos mostrados en el combo.
- El aspecto visual del componente es el siguiente:



3. INSTALACIÓN Y CONFIGURACIÓN

En este apartado se incluye información sobre la instalación y la configuración del componente combo de Valores.

3.1. INSTALACIÓN

El componente combo de valores ya viene instalado en el arquetipo Web, incluido con el módulo de componentes visuales. Por este motivo no es necesaria una instalación adicional si se parte del arquetipo.

3.2. CONFIGURACIÓN

Para la configuración del componente se debe crear una sentencia de consulta para el combo.

Este paso es necesario para que el componente tenga acceso a los datos de la base de datos que queremos que sean mostrados en el combo.

Tenemos que establecer en el fichero queries.properties de la aplicación para cada combo de valores una consulta SQL para obtener el listado. El nombre del parámetro debe componerse de la siguiente manera y ser único en el fichero de propiedades:

queries.properties

```
queryCode.nombreQuery=SELECT CAMPO1, CAMPO2 .... FROM ... WHERE ...
```

El texto en color rojo es el identificador de la consulta. Es el valor que es necesario poner en el atributo *queryCode* cuando se establece un elemento *comboValores* en la página, por el cuál se puede seleccionar esta consulta cuando se inserta la etiqueta de este componente en un formulario.

4. USO

Una vez instalado el módulo de componentes puede procederse a su utilización. Para usar el componente es necesario seguir los pasos descritos en los siguientes apartados:

4.1. Paso 1: Definición de la consulta del combo de valores

Es necesario especificar en el fichero queries.properties de la aplicación la consulta que tiene que usarse por el combo de valores que pongamos en nuestro formulario. Por ejemplo:

queries.properties

```
# Queries para los ejemplos de combo de valores

queryCode.provinciasCombo=SELECT CDPROV, DSPROV, CDPAIS, CDCOMU FROM
SUCA_PROVINCIA

queryCode.localidadesCombo=SELECT CDMUNI, DSMUNI, CDPAIS, CDPROV FROM
SUCA_MUNICIPIO
```

En este ejemplo se definen dos consultas para dos ejemplos de combo de valores, una para provincias y otra para localidades.

4.2. Paso 2: Declaración en el bean de respaldo de la página de un objeto de tipo atlas.componente.bo.ComboValores e inicialización del mismo.

El bean de respaldo de la página debe definirse en el fichero de configuración de JSF que puede encontrarse en el proyecto web generado por el arquetipo bajo este nombre: *\src\main\webapp\WEB-INF\faces-managed-beans.xml.*

faces-managed-beans.xml

```
<managed-bean>
    <description>Bean de respaldo de los ejemplos de combo de
valores</description>
    <managed-bean-name>comboValoresSampleBean</managed-bean-name>
    <managed-bean-class>
        atlas.samples.jsf.ComboValoresSampleBean
    </managed-bean-class>
    <managed-bean-scope>request</managed-bean-scope>
    <managed-property>
      <property-name>facade</property-name>
      <value>#{EjplFacade}</value>
    </managed-property>
</managed-bean>
```

En dicho bean debe crearse un objeto de tipo atlas.componente.bo.ComboValores e inicializarlo. Por cada combo de valores será necesario crear un objeto de este tipo.

ComboValoresSampleBean.java

```
/**
 * The Class ComboValoresSampleBean.
 */
public class ComboValoresSampleBean {
    ...

    /** Servicio de la fachada */
    private EjplFacade facade;


    /** Combo de valores de municipios. */
    private ComboValores comboValoresProvincias;
    /** Combo de valores de municipios. */
    private ComboValores comboValoresMunicipios;

    /** Constructor sin parámetros */
    public ComboValoresSampleBean() {
        this.comboValoresProvincias = new ComboValores();
        this.comboValoresMunicipios = new ComboValores();
    }

...
```

```
    /**
     * @return El combo de valores de municipios.
     */
    public ComboValores getComboValoresMunicipios() {
        return comboValoresMunicipios;
    }
    /**
     * Establece el combo de valores de municipios.
     * @param comboValoresMunicipios El combo de valores
     */
    public void setComboValoresMunicipios(ComboValores
        comboValoresMunicipios) {
        this.comboValoresMunicipios = comboValoresMunicipios;
    }

    /**
     * Estable el combo de valores de provincias
     * @param comboValoresProvincias El combo de valores
     */
    public void setComboValoresProvincias(ComboValores
comboValoresProvincias) {
        this.comboValoresProvincias = comboValoresProvincias;
    }

    /** @return El combo de valores de provincias  */
    public ComboValores getComboValoresProvincias() {
        return comboValoresProvincias;
    }

    /**  @return la fachada */
    public EjplFacade getFacade() {
        return facade;
    }

    /** @param facade. * Asocia la fachada al Backing Bean   */
    public void setFacade(EjplFacade facade) {
        this.facade = facade;
    }
   ...
}
```

Nota
Si la pantalla jsf que se está realizando ya tiene asociado un Bean no hace falta hacer otro. Se puede añadir a este Bean existente la variables privadas con sus correspondientes getter y setter. Además de inicializarlas en el constructor del bean.

4.3. Paso 3: Definición del espacio de nombres de etiquetas de Atlas

Es necesario crear un fichero *xhtml* y establecer la definición del espacio de nombres para las etiquetas de componentes de Atlas. Un ejemplo de cabecera de fichero *xhtml* es la siguiente:

Cabecera de fichero xhtml

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:atlas="http://atlas.core.componentes/jsf"
      xmlns:a4j="http://richfaces.org/a4j">
```

4.4. Paso 4: Inserción en la página de la etiqueta atlas:comboValores

Es necesario insertar la etiqueta comboValores en el formulario de una página.

A continuación se muestra el ejemplo propuesto anteriormente, provincias y municipios:

Ejemplo.xhtml

```
<h:outputLabel for="comboValoresProvincias" styleClass="label"
    value="#{bundle['lv.provincia']}" />
<atlas:comboValores id="comboValoresProvincias"
    queryCode="provinciasCombo" order="DSPROV"
    value="#{comboValoresSampleBean.valueProvincia}"
    dataModel="#{comboValoresSampleBean.comboValoresProvincias}"
    filtro="CDPAIS LIKE '%ES%'" render="comboValoresMunicipios"
    registros="60" valuePattern="%1-%2"
    valueChangedListener="comboValoresSampleBean.valueChanged"/>

<h:outputLabel for="comboValoresMunicipios" styleClass="label"
    value="#{bundle['lv.localidad']}" />
<atlas:comboValores id="comboValoresMunicipios"
    queryCode="localidadesCombo" order="DSMUNI"
    value="#{comboValoresSampleBean.valueLocalidad}"
    dataModel="#{comboValoresSampleBean.comboValoresMunicipios}"
    filtro="CDPAIS LIKE '%ES%' AND CDPROV
='#{comboValoresSampleBean.codigoProvincia}'"
    ajaxUpdate="false" valuePattern="%1-%2"/>
```

4.5. Atributos del componente.

Los principales atributos del componente atlas:comboValores son los siguientes:

- El atributo *id* sirve para establecer el identificador del componente. Es obligatorio y debe ser único.
- El atributo *dataModel* sirve para enlazar el componente con el objeto de la clase atlas.componentes.bo.ComboValores declarado.
- El atributo *queryCode* sirve para especificar cuál es la consulta a base de datos que se quiere utilizar para obtener los elementos que el combo va a mostrar.
- El atributo *filtro* sirve para establecer un filtro SQL a la consulta especificada con el atributo *queryCode*. Si el filtro contiene más de una clausula se debe añadir el AND entre ambas, pero no al principio. En el filtro se pueden añadir enlaces a variables de un backing bean, de modo que la query sea dinámica, en la caso anterior el filtro de municipio se enlaza con el código de la provincia recogida en el combo de provincias.
- El atributo *order* permite especificar la columna de la consulta por la que se quiere ordenar.
- El campo *ajaxUpdate* especifica si se debe actualizar el valor al seleccionarlo o no.
- El atributo *render* contendrá una lista de identificadores de los componentes que tengan que actualizarse al seleccionar un elemento del combo. Sólo es aplicable si *ajaxUpdate* es *true*.

La lista completa de atributos del componente comboValores son los siguientes:

Nombre atributo	Obligatorio	Descripción
Id	SI	Identificador del componente. Es obligatorio y debe ser único.
dataModel	SI	Clase que se encarga de controlar el comportamiento del combo. Se trata de un objeto de la clase atlas.componentes.bo.ComboValores que se encuentra declarado e inicializado en un bean de respaldo.
queryCode	SI	Código de la consulta SQL necesaria para obtener los datos que se van a mostrar, debe corresponderse con el distintivo de una propiedad en el fichero *queries.properties* con formato queryCode.valorDeEstaPropiedad.
filtro	NO	Filtro a añadir a la consulta indicada en queryCode.
order	NO	Columna de ordenación. Debe estar contenida en la clausula select de la query referenciada en la propiedad *queryCode*.
registros	NO	Número máximos de registros que se mostrarán, si no se especifican se mostrarán todos los que devuelva la consulta.

valuePattern	NO	Patrón para el atributo value del combo. Si no se especifica se utilizará la primera columna de la clausula select. Se compone con texto intercalando %i, donde i es la posición de la columna en la consulta. Ejemplo: "%1 - %2 - %3".
labelPattern	NO	Patrón para la información que se va a mostrar al usuario. Si no se especifica se mostrará la segunda columna de la clausula select. Se compone con texto intercalando %i, donde i es la posición de la columna en la consulta. Ejemplo: "%1 - %2 - %3".
externalProvider	NO	Permite enlazar al componente una lista externa de resultados. Debe ser una expresión EL que devuelva una lista de objetos SelectItem. Si se utiliza este atributo no se tendrán en cuenta la consulta especificada en *queryCode* ni el resto de atributos relacionados: *filtro, order, registros, valuePattern, labelPatter.*
required	NO	Valor true o false para indicar si es obligatorio que el usuario seleccione un elemento. El valor por defecto es false.
requiredMessage	NO	Texto a utilizar para el mensaje en caso en que sea obligatorio seleccionar un elemento y no se seleccione.
rendered	NO	Valor true o false para especificar si debe renderizarse el componente. Su valor por defecto es true.
value	NO	Expresión EL con el valor del backing bean que va a recibir el valor seleccionado en el componente.
enableManualInput	NO	Valor true o false para especificar si se permite escribir sobre el combo para filtrar elementos del componente. Su valor por defecto es true.
selectFirst	NO	Valor true o false. Si es true cuando el usuario escriba sobre el campo se seleccionará el primer registro resultado. Valor por defecto true. Sólo es aplicable si la propiedad *enableManualInput* es true.
ajaxUpdate	NO	Valor true o false para especificar si debe realizarse una petición ajax al seleccionar un elemento del componente.
execute	NO	Lista de identificadores de componentes que se procesarán en una petición ajax. Por defecto se procesa el propio componente.
render	NO	Lista de identificadores de componentes que se renderizarán en una petición ajax.
disabled	NO	Valor true o false, si es true el componente está deshabilitado. Valor por defecto false.
immediate	NO	Valor true o false. Si es true el componente se salta la fase de validación en una petición. Valor por defecto false.
inputWidth	NO	Tamaño en pixels del campo input del componente. Su valor por defecto es 182.
style	NO	Estilos CSS para el componente.

styleClass	NO	Clase/s CSS para el componente.
itemClass	NO	Clase/s CSS para los elementos de la lista.
listClass	NO	Clase/s CSS para la lista.
listHeight	NO	Altura de la lista.
listWidth	NO	Anchura de la lista.
maxListHeight	NO	Altura máxima para la lista.
minListHeight	NO	Altura mínima para la lista.
selectItemClass	NO	Clase/s CSS a aplicar al elemento seleccionado en la lista.
showButton	NO	Valor true o false. Si es true se muestra el icono para expandir la lista. Valor por defecto true.
tabindex	NO	Propiedad tabindex HTML para especificar el orden de tabulación del componente.
defaultLabel	NO	Etiqueta por defecto para el componente cuando no hay ningún elemento seleccionado.
onblur	NO	Código javascript ejecutado cuando el componente pierde el foco.
onchange	NO	Código javascript ejecutado cuando el componente pierde el foco y su valor se ha modificado desde que tiene el foco.
onclick	NO	Código javascript ejecutado al hacer click sobre el componente.
ondblclick	NO	Código javascript ejecutado al hacer doble click sobre el componente.
onmousedown	NO	Código javascript que se ejecuta al presionar el ratón sobre el componente
onmousemove	NO	Código javascript que se ejecuta cuando se mueve el puntero del ratón dentro de este componente.
onmouseout	NO	Código javascript que se ejecuta cuando se mueve el ratón situado encima de este componente hacia fuera del mismo.
onmouseover	NO	Código javascript que se ejecuta cuando el puntero del ratón se mueve desde fuera del componente hasta situarse encima del mismo.
onmouseup	NO	Código javascript que se ejecuta cuando se suelta sobre el componente un botón del ratón que se mantenía presionado.
onfocus	NO	Código javascript que se ejecuta cuando el componente recibe el foco.

onkeydown	NO	Código javascript que se ejecuta cuando se pulsa una tecla sobre el componente.
onkeypress	NO	Código javascript que se ejecuta cuando se pulsa y se deja de pulsar una tecla sobre el componente.
onkeyup	NO	Código javascript que se ejecuta cuando se deja de pulsar una tecla sobre el componente.
onlistclick	NO	Código javascript que se ejecuta al hacer click sobre la lista del componente.
onlistdblclick	NO	Código javascript que se ejecuta al hacer doble click sobre la lista del componente.
onlisthide	NO	Código javascript que se ejecuta cuando se oculta la lista del componente
onlistkeydown	NO	Código javascript que se ejecuta cuando se pulsa una tecla sobre la lista del componente.
onlistkeypress	NO	Código javascript que se ejecuta cuando se pulsa y se deja de pulsar una tecla sobre la lista del componente.
onlistkeyup	NO	Código javascript que se ejecuta cuando se deja de pulsar una tecla sobre la lista del componente.
onlistmousedown	NO	Código javascript que se ejecuta al presionar el ratón sobre la lista del componente
onlistmousemove	NO	Código javascript que se ejecuta cuando se mueve el puntero del ratón dentro de la lista del componente.
onlistmouseout	NO	Código javascript que se ejecuta cuando se mueve el ratón situado encima de la lista del componente hacia fuera de la misma.
onlistmouseover	NO	Código javascript que se ejecuta cuando el puntero del ratón se mueve desde fuera de la lista del componente hasta situarse encima del mismo.
onlistmouseup	NO	Código javascript que se ejecuta cuando se suelta sobre la lista del componente un botón del ratón que se mantenía presionado.
onlistshow	NO	Código javascript que se ejecuta cuando se muestra la lista del componente.
onselectitem	NO	Código javascript que se ejecuta cuando se selecciona un elemento del componente.
converter	NO	Conversor JSF para el componente.
converterMessage	NO	Mensaje para mostrar si falla la conversión.
title	NO	Título para el componente.

4.6. Etiquetas y valores del combo.

Con el atributo *value* se especifica la variable del backing bean a la que se quiere enlazar el valor seleccionado en el combo.
Por defecto el combo selecciona el primer elemento de la clausula select de la consulta como value, es decir, como el valor que enviará al backing bean, y el segundo elemento como label., es decir, como el texto que se mostrará en el combo.

queries.properties

```
# Queries para los ejemplos de combo de valores

queryCode.provinciasCombo=SELECT CDPROV, DSPROV, CDPAIS, CDCOMU FROM
SUCA_PROVINCIA
```

En el ejemplo anterior CDPROV sería el value y DSPROV el label del combo.
Sin embargo, quizá necesitemos enviar más de un valor al backing bean o mostrar más de un valor en pantalla. Para que esto no implique rehacer la consulta se pueden utilizar los atributos *valuePattern* y *labelPattern*, En esos atributos se puede definir una composición de los diferentes elementos que forman la clausula select, referenciando cada elemento con el símbolo % seguido por el número de orden del campo en la consulta empezando por 1. Por ejemplo, el valor *valuePattern="%1-%2"* establecería como value para el combo la composición *CDPROV-DSPROV*. Del mismo modo con *labelPattern* para la información que se muestra al usuario,

5. RECOMENDACIONES Y BUENAS PRÁCTICAS

Se recomienda restringir la consulta o utilizar el campo registros de modo que el número de elemento no sea excesivo, ya que puede ralentizar el funcionamiento de la aplicación.

5.1. EJEMPLO DE USO

Se puede ver varios ejemplos de dicho componente en la siguiente secuencia de navegación por la aplicación de demostración de los componentes:

Inicio > Formularios > Otros componentes Atlas -> Combo de valores

6. PREGUNTAS MÁS FRECUENTES

En este apartado se incluyen una lista de preguntas más frecuentes sobre el componente.

Pregunta: ¿Dónde puedo encontrar información general sobre los componentes?

Respuesta: En la aplicación de demostración de los componentes del Framework Atlas

Pregunta: ¿Cómo puedo encadenar combos de valores? (que uno dependa de lo seleccionado en otro)

Respuesta: Usando un valor de retorno del combo padre en el atributo filtro del combo hijo. Ejemplo:

Ejemplo.xhtml

```
<h:outputLabel for="comboValoresProvincias" styleClass="label"
    value="#{bundle['lv.provincia']}" />
<atlas:comboValores id="comboValoresProvincias"
    queryCode="provinciasCombo" order="DSPROV"
    value="#{comboValoresSampleBean.codigoProvincia}"
    dataModel="#{comboValoresSampleBean.comboValoresProvincias}"
    filtro="CDPAIS LIKE '%ES%'" render="comboValoresMunicipios"
    registros="60" valuePattern="%1-%2"
    valueChangedListener="comboValoresSampleBean.valueChanged"/>

<h:outputLabel for="comboValoresMunicipios" styleClass="label"
    value="#{bundle['lv.localidad']}" />
<atlas:comboValores id="comboValoresMunicipios"
    queryCode="localidadesCombo" order="DSMUNI"
    value="#{comboValoresSampleBean.codigoLocalidad}"
    dataModel="#{comboValoresSampleBean.comboValoresMunicipios}"
    filtro="CDPAIS LIKE '%ES%' AND CDPROV
='#{comboValoresSampleBean.codigoProvincia}'"
    ajaxUpdate="false" valuePattern="%1-%2"/>
```

7. ENLACES RELACIONADOS

Producto	URL
Ajax4JSF	http://www.jboss.org/jbossrichfaces/
Barbecue	http://barbecue.sourceforge.net/
Commons BeanUtils	commons.apache.org/beanutils/
Commons Configurations	http://commons.apache.org/configuration/
Facelets	https://facelets.dev.java.net/
Hibernate	http://www.hibernate.org/
Hibernate Annotations	http://www.hibernate.org/hib_docs/annotations/reference/en/html_single/
JAXB	http://java.sun.com/webservices/jaxb/
JPA	http://java.sun.com/developer/technicalArticles/J2EE/jpa/
JSF	http://java.sun.com/javaee/javaserverfaces/
JSFUnit	http://www.jboss.org/jsfunit/
Log4J	http://logging.apache.org/log4j/
MyFaces Core	http://myfaces.apache.org/
RichFaces	http://www.jboss.org/jbossrichfaces/
Spring	http://www.springframework.org/
Spring Security	http://www.springframework.org/
Velocity	http://velocity.apache.org/